

بسم الله الرحمن الرحيم



شبکه های مخابرات داده

دکتر پاکروان

گزارش تمرین سری 3

امیرحسین رستمی (96101635)

دانشگاه صنعتی شریف

توضیحاتی راجع به فایل های ضمیمه شده:

پس از بازکردن فایل ضمیمه شده 6 فایل را مشاهده خواهید کرد:

0- فایل test.cc

1- فولدر Codes&Data

2- فولدر teminalPics

3- فایل متلب graphs.mlx

4- فایل pdf گزارش (به ذکر این مورد نمی پردازیم چون بدیهی است که چه چیزی است)

5- فایل word گزارش (به ذکر این مورد نمی پردازیم چون بدیهی است که چه چیزی است)

6- فایل HW3.cc

فایل test.cc: این فایل، قطعه کد بخش getting started with NS3 است که در ابتدای تمرین مطرح گردیده است.

فولدر Codes&Data: شامل 4 (non persistent, p=0.3 persistent, aloha, p=0.6 persistent) فولدر است که در هر کدام خروجی های tr. و pcap. و animation برای هر 7 دیتا ریت پرسیده شده آورده شده است، **هم چنین در فولدر های p persistent و aloha هم کد تغییر یافته ی csma.cc موجود در کتابخانه NS3 هم آورده شده است.**

فولدر teminalPics: شامل 4 (non persistent, p=0.3 persistent, aloha, p=0.6 persistent) فولدر است که در هر کدام تصاویری که به صورت اسکرین شات از خروجی ترمینال در هر تست است وجود دارد.

فایل متلب graphs.mlx: کد متلب زده شده جهت ترسیم نمودار های حاصله از اطلاعات به دست آمده از تست های انجام شده روی شبکه.

فایل HW3.cc: کد هسته ی تمرین که به طراحی شبکه، پردازش throughput و delay، فراخوانی الگوریتم های لایه ی مک و تولید و ذخیره سازی فایل های xml, .pcap, .tr. می پردازد.

حل بخش getting familiar with NS3:

کد زده شده:

```
1 #include "ns3/core-module.h"
2
3
4 using namespace ns3 ;
5 NS_LOG_COMPONENT_DEFINE("Lab#3");
6
7 int main (int argc , char * argv[]) {
8     int a ;
9     int b ;
10    CommandLine myCMD;
11    myCMD.AddValue("a", "This is a", a ) ;
12    myCMD.AddValue("b", "This is b", b ) ;
13    myCMD.Parse (argc , argv ) ;
14    std::cout << " a + b = " << a + b ;
15    return 0;
16 }
17
```

خروجی ترمینال:(چندین بار ران کردم)

```
[2619/2668] Compiling scratch/Test.cc
[2628/2668] Linking build/scratch/Test
Waf: Leaving directory '/home/amirhosein/ns3/ns-allinone-3.30/ns-3.30/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (2.449s)
a + b = 7(base) amirhosein@amirhosein-UX550VD:~/ns3/ns-allinone-3.30/ns-3.30$ .
"scratch/Test --a=2 --b=5"
Waf: Entering directory '/home/amirhosein/ns3/ns-allinone-3.30/ns-3.30/build'
Waf: Leaving directory '/home/amirhosein/ns3/ns-allinone-3.30/ns-3.30/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.723s)
a + b = 7(base) amirhosein@amirhosein-UX550VD:~/ns3/ns-allinone-3.30/ns-3.30$ .
"scratch/Test --a=2 --b=5"
Waf: Entering directory '/home/amirhosein/ns3/ns-allinone-3.30/ns-3.30/build'
Waf: Leaving directory '/home/amirhosein/ns3/ns-allinone-3.30/ns-3.30/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.735s)
a + b = 7(base) amirhosein@amirhosein-UX550VD:~/ns3/ns-allinone-3.30/ns-3.30$ .
"scratch/Test --a=2 --b=5"
Waf: Entering directory '/home/amirhosein/ns3/ns-allinone-3.30/ns-3.30/build'
Waf: Leaving directory '/home/amirhosein/ns3/ns-allinone-3.30/ns-3.30/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.733s)
a + b = 7(base) amirhosein@amirhosein-UX550VD:~/ns3/ns-allinone-3.30/ns-3.30$
```

همانطور که ملاحظه می کنید خروجی مطلوب و مورد انتظار به دست آمده است.

توضیحات اولیه راجع به کد

از آنجا که در فایل آموزش NS3 شرح دقیقی از کلاس های لایبری NS3 شده است **لذا ما از توضیح کلاس های مشابه با آن که در کد استفاده شده است پرهیز می کنیم و مستقیم سراغ خطوط جدیدی که در کد زده شده است می رویم.**

```
CsmaHelper csma;  
csma.SetChannelAttribute ("DataRate", StringValue ("1024Kbps"));  
csma.SetChannelAttribute ("Delay", StringValue ("2ms"));  
NetDeviceContainer devices = csma.Install (node);
```

خط اول تا پایان سوم: خط ابتدایی یک کلاس کمک کننده ی توپولوژی است که csma Helper نامیده می شود تا کانال و دیوایس شبکه کشیده شود. سرعت دیتای دیوایس شبکه 1024 کیلو بیت بر ثانیه است و تأخیر انتقال کانال 2 میلی ثانیه است.

خط چهارم: این کد کشیدن شبکه و کانال را به اتمام می رساند. از یک توپولوژی کمک کننده به نام NetDeviceContainer استفاده می شود تا شیء های net device ساخته شوند. این همانند کاری است که NodeContainer برای مدیریت اشیاء نود انجام می داد.

```
InternetStackHelper internet;  
internet.Install (node);  
  
Ipv4AddressHelper address;  
address.SetBase ("11.63.5.0", "255.255.255.0");  
Ipv4InterfaceContainer interfaces = address.Assign (devices);
```

دو خط اول: پس از setup نود ها حال باید پروتکل استک را روی آن ها نصب کنیم که توسط دو خط اول انجام می گردد. اینترنت استک پروتکل هایی از جمله TCP, UDP, IP و غیره را فعال می کند. توسط این کلاس مسیریابی بسته ها توسط لایه network انجام می شود که در این تمرین با آدرس IP انجام خواهد شد.

پیش از بررسی ادامه خطوط ابتدا شماره دانشجویی خود را تشریح می کنیم:

96101635 = XXXXstnu -> s = 1, t = 6, n = 3, u = 5 -> setBase("11.63.5.0", "255.255.255.0");

خطوط سوم تا پنجم: این کدها با آدرس نودها سروکار دارد. به هر نود یک آدرس IP اختصاصی داده می شود. طبق گفته ی گزارش کار و با توجه به شماره دانشجویی خود IP ها را تعیین می کنیم. آدرس ها به صورت 11.63.5.1 به نود اول، 11.63.5.2 به نود دوم و به همین ترتیب به نودها اختصاص می یابند. در خط آخر آدرس واقعی به هر گره اختصاص پیدا می کند. در حقیقت آدرس IP با کارت شبکه با شیء Ipv4Interface پیوند داده می شود.

در ادامه کد تا رسیدن به خط tracing، مشابه توتوریال سایت جهت برقراری اتصالات نود های شبکه (سرور، کلاینت) است، لذا چون مشابه توتوریال است از ذکر آن پرهیز می کنیم و سراغ خط tracing به بعد می رویم.

```
// Tracing
AsciiTraceHelper ascii;
csma.EnableAscii(ascii.CreateFileStream ("HW3_non.tr"), devices);
csma.EnablePcap("HW3_non", devices, false);
```

با کمک این سه خط فایل trace و pcap ساخته می شوند. (این نمونه کد برای حالت non persistent است).

```
// Calculate Throughput using Flowmonitor
FlowMonitorHelper myflow;
Ptr<FlowMonitor> monitor = myflow.InstallAll();
```

می دانیم که در محاسبه ی throughput & delay نیاز به مانیتورینگ flow جاری در شبکه داریم لذا به کمک دو خط کد فوق به tracking این امور می پردازیم.

```
MobilityHelper mobility;
mobility.SetPositionAllocator ("ns3::GridPositionAllocator",
    "MinX", DoubleValue (0.0),
    "MinY", DoubleValue (0.0),
    "DeltaX", DoubleValue (10.0),
    "DeltaY", DoubleValue (10.0),
    "GridWidth", UIntegerValue (5),
    "LayoutType", StringValue ("RowFirst"));

mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
mobility.Install (node);

AnimationInterface anim ("animation.xml");
anim.SetConstantPosition (node.Get (0), 0.0, 20.0);
anim.SetConstantPosition (node.Get (1), 5.0, 20.0);
anim.SetConstantPosition (node.Get (2), 10.0, 20.0);
anim.SetConstantPosition (node.Get (3), 15.0, 20.0);
anim.SetConstantPosition (node.Get (4), 20.0, 20.0);
anim.SetConstantPosition (node.Get (5), 25.0, 20.0);
anim.SetConstantPosition (node.Get (6), 30.0, 20.0);
anim.SetConstantPosition (node.Get (7), 35.0, 20.0);
```

به کمک قطعه خط کد فوق به طراحی فایل انیمیشن شبکه می پردازیم، طبق داکيومنت های موجود در اینترنت دریافتیم که کلاس MobilityHelper از design pattern عه Builder تبعیت می کند و لذا پس از تعریف پوینتری از جنس MobilityHelper به build کردن آن (که به کمک دستور install در اینجا صورت میگیرد) می پردازیم، در ادامه محل ذخیره شدن interface گرافیکی ساخته شده را تعیین می کنیم. (مختصات های داده شده کاملاً دلخواه اند و به این دلیل y ها را برابر دادیم چون خواستیم طبق گفته صورت تمرین نود ها به صورت یک bus افقی باشند).

```
Ptr<Ipv4FlowClassifier> classifier = DynamicCast<Ipv4FlowClassifier> (myflow.GetClassifier ());  
std::map<FlowId, FlowMonitor::FlowStats> stats = monitor->GetFlowStats ();
```

برای دسترسی به اطلاعات flow موجود در شبکه به کمک classifier، اطلاعات را از FlowMonitor استخراج می کنیم و هم چنین به کمک خط بعدی status موجود در شبکه را که یک دیکشنری با کلید های شماره ی Flow و FlowStats Value است به دست می آوریم تا به کمک آن ها در ادامه tracking کنیم فلوی موجود در شبکه را.

نکته ی مهم: خروجی هر findFlow از Classifier یک Tuple (این ساختار یک داده ساختار معروف در پایتون است که تقریباً به معنای لیست ثابت است) است که 5 عنصر داشته و آن ها عبارت اند از:

sourceAddress-1

destinationAddress-2

IP protocol number-3

nextLayer Source IP-4

nextLayer Destination IP-5

سپس به کمک اطلاعات موجود در این فیلدها به بررسی اطلاعات ترافیک موجود در شبکه می پردازیم.

نحوه ی محاسبه ی Delay و Throughput هم واضحاً در کد ضمیمه شده موجود است و از ذکر جزئیات بیشتر (برای جلوگیری از زیاد شدن حجم گزارش پرهیز می کنیم).

این کانال بسته ای با طول بیش از 4400 بایت از خود عبور نمی دهد. بنابراین برای اینکه تمام این سرعت ها را بتوان پیاده سازی کرد نیاز است اینتروال را 0.034 قرار داد تا ماکسیمم این بسته ها از 4400 بیشتر نشود. از آنجایی که تنها 1 بسته فرستاده می شود مقدار اینتروال هیچ تأثیری در مقادیر دلی و تروپوت ندارند و هم چنین چون می خواهیم مقایسه های الگوریتم های مختلف لایه مک رو با هم انجام دهیم باید interval یکسانی را برای همه ی ورودی ها قرار داد تا نتایج حقیقی مشخص و قابل مقایسه با یکدیگر شود.

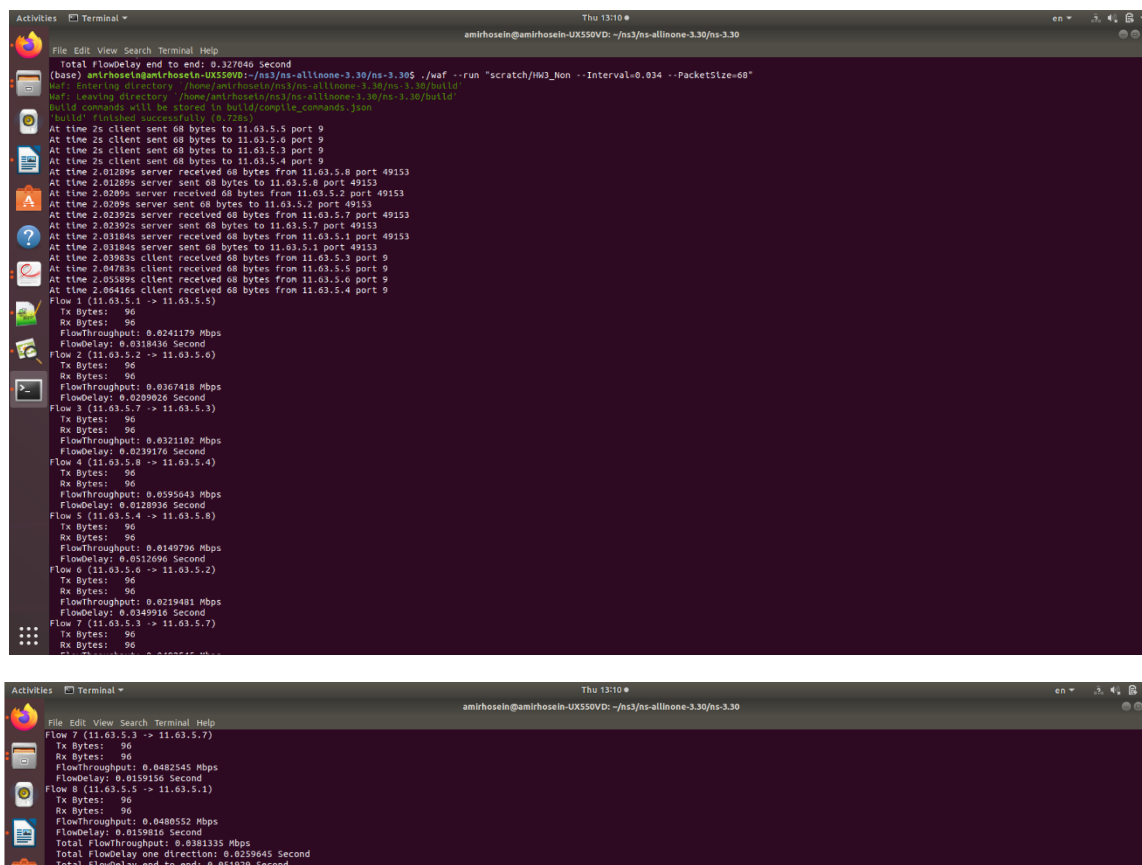
در ادامه به بررسی خروجی های الگوریتم های non-persistent ، p-persistent و ALOHA می پردازیم.

:Non persistent

از آنجاکه ما الگوریتم را به ازای 7 دیتا ریت مختلف محاسبه می کنیم، داریم که اگر بخواهیم خروجی ترمینال را در هر قسمت بیاوریم، خیلی حجم گزارش زیاد می شود و لذا فقط خروجی ترمینال را در حالت non persistent می آوریم و در بخش های بعدی با اجازه آقای مهندس اخوان، بنده فقط نمودارهای جامع حاوی همه دیتا ریت ها (به کمک نرم افزار متلب) را می آورم.

(توجه کنید که تمامی تصاویر ترمینال تمامی قسمت ها در پاسخ آپلودی آورده شده است)

1- دیتا ریت 16 کیلوبیت بر ثانیه:



```
amirhosein@amirhosein-UX550VD: ~/ns3/ns-allinone-3.30/ns-3.30
Total FlowDelay end to end: 0.327046 Second
(base) amirhosein@amirhosein-UX550VD:~/ns3/ns-allinone-3.30/ns-3.30$ ./waf --run "scratch/HW3_Non --Interval=0.034 --PacketSize=60"
waf: Entering directory `./'
waf: Leaving directory `./'
Build commands will be stored in build/compile_commands.json
BuildID: 7216889 successfully (0.728s)
At time 2s client sent 60 bytes to 11.63.5.5 port 9
At time 2s client sent 60 bytes to 11.63.5.6 port 9
At time 2s client sent 60 bytes to 11.63.5.3 port 9
At time 2s client sent 60 bytes to 11.63.5.4 port 9
At time 2.01289s server received 60 bytes from 11.63.5.8 port 49153
At time 2.01289s server sent 60 bytes to 11.63.5.8 port 49153
At time 2.0289s server received 60 bytes from 11.63.5.2 port 49153
At time 2.0289s server sent 60 bytes to 11.63.5.2 port 49153
At time 2.0292s server received 60 bytes from 11.63.5.7 port 49153
At time 2.0292s server sent 60 bytes to 11.63.5.7 port 49153
At time 2.03184s server received 60 bytes from 11.63.5.1 port 49153
At time 2.03184s server sent 60 bytes to 11.63.5.1 port 49153
At time 2.03983s client received 60 bytes from 11.63.5.3 port 9
At time 2.04783s client received 60 bytes from 11.63.5.5 port 9
At time 2.05589s client received 60 bytes from 11.63.5.6 port 9
At time 2.0646s client received 60 bytes from 11.63.5.4 port 9
Flow 1 (11.63.5.1 -> 11.63.5.5)
Tx Bytes: 90
Rx Bytes: 90
FlowThroughput: 0.0241179 Mbps
FlowDelay: 0.0318436 Second
Flow 2 (11.63.5.2 -> 11.63.5.6)
Tx Bytes: 90
Rx Bytes: 90
FlowThroughput: 0.0367418 Mbps
FlowDelay: 0.0290826 Second
Flow 3 (11.63.5.7 -> 11.63.5.3)
Tx Bytes: 90
Rx Bytes: 90
FlowThroughput: 0.0321102 Mbps
FlowDelay: 0.0239170 Second
Flow 4 (11.63.5.8 -> 11.63.5.4)
Tx Bytes: 90
Rx Bytes: 90
FlowThroughput: 0.0595043 Mbps
FlowDelay: 0.0128936 Second
Flow 5 (11.63.5.4 -> 11.63.5.8)
Tx Bytes: 90
Rx Bytes: 90
FlowThroughput: 0.0140796 Mbps
FlowDelay: 0.0312694 Second
Flow 6 (11.63.5.6 -> 11.63.5.2)
Tx Bytes: 90
Rx Bytes: 90
FlowThroughput: 0.0219481 Mbps
FlowDelay: 0.0349910 Second
Flow 7 (11.63.5.3 -> 11.63.5.7)
Tx Bytes: 90
Rx Bytes: 90
FlowThroughput: 0.0480552 Mbps
FlowDelay: 0.0159816 Second
Total FlowThroughput: 0.0381335 Mbps
Total FlowDelay one direction: 0.0259645 Second
Total FlowDelay end to end: 0.051929 Second
```

Total Throughput: 0.038135 Mbps

Total end to end delay: 0.051929 seconds

2- دیتا ریت 32 کیلو بیت بر ثانیه: (صرفا بخش انتهایی خروجی ترمینال را که حاوی مقادیر مدنظر ما است آورده ام)

```
amirhosein@amirhosein-UXSS50VD: ~/ns3/ns-allinnone-3.30/ns-3.30
Flow 4 (11.63.5.8 -> 11.63.5.4)
Tx Bytes: 164
Rx Bytes: 164
FlowThroughput: 0.897729 Mbps
FlowDelay: 0.0134249 Second
Flow 5 (11.63.5.4 -> 11.63.5.8)
Tx Bytes: 164
Rx Bytes: 164
FlowThroughput: 0.8621923 Mbps
FlowDelay: 0.0210959 Second
Flow 6 (11.63.5.5 -> 11.63.5.1)
Tx Bytes: 164
Rx Bytes: 164
FlowThroughput: 0.844104 Mbps
FlowDelay: 0.0297479 Second
Flow 7 (11.63.5.3 -> 11.63.5.7)
Tx Bytes: 164
Rx Bytes: 164
FlowThroughput: 0.8754463 Mbps
FlowDelay: 0.0173439 Second
Flow 8 (11.63.5.6 -> 11.63.5.2)
Tx Bytes: 164
Rx Bytes: 164
FlowThroughput: 0.875299 Mbps
FlowDelay: 0.0174239 Second
Total FlowThroughput: 0.8576196 Mbps
Total FlowDelay one direction: 0.0258442 Second
Total FlowDelay end to end: 0.0516885 Second
(base) amirhosein@amirhosein-UXSS50VD:~/ns3/ns-allinnone-3.30/ns-3.30$ ./waf --run "scratch/HW3_Non --Interval=0.034 --PacketSize=272"
waf: Entering directory '/home/amirhosein/ns3/ns-allinnone-3.30/ns-3.30/build'
waf: leaving directory '/home/amirhosein/ns3/ns-allinnone-3.30/ns-3.30/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.736s)
At time 2s client sent 272 bytes to 11.63.5.5 port 9
At time 2s client sent 272 bytes to 11.63.5.6 port 9
At time 2s client sent 272 bytes to 11.63.5.3 port 9
At time 2s client sent 272 bytes to 11.63.5.4 port 9
At time 2.01449s server received 272 bytes from 11.63.5.8 port 49153
At time 2.01449s server sent 272 bytes to 11.63.5.8 port 49153
At time 2.01927s server received 272 bytes from 11.63.5.7 port 49153
At time 2.01927s server sent 272 bytes to 11.63.5.7 port 49153
At time 2.02876s server received 272 bytes from 11.63.5.2 port 49153
At time 2.02876s server sent 272 bytes to 11.63.5.2 port 49153
At time 2.03844s server received 272 bytes from 11.63.5.1 port 49153
At time 2.03844s server sent 272 bytes to 11.63.5.1 port 49153
At time 2.04895s client received 272 bytes from 11.63.5.6 port 9
At time 2.0578s client received 272 bytes from 11.63.5.3 port 9
At time 2.06735s client received 272 bytes from 11.63.5.4 port 9
At time 2.07684s client received 272 bytes from 11.63.5.5 port 9
Flow 1 (11.63.5.1 -> 11.63.5.5)
Tx Bytes: 300
Rx Bytes: 300
FlowThroughput: 0.8024395 Mbps
FlowDelay: 0.0384434 Second
Flow 2 (11.63.5.2 -> 11.63.5.6)
```

Total Throughput: 0.0576196 Mbps

Total end to end delay: 0.0516885 seconds

3- دیتا ریت 64 کیلو بیت بر ثانیه:

```
amirhosein@amirhosein-UXSS50VD: ~/ns3/ns-allinnone-3.30/ns-3.30
At time 2.01927s server received 272 bytes from 11.63.5.7 port 49153
At time 2.01927s server sent 272 bytes to 11.63.5.7 port 49153
At time 2.02876s server received 272 bytes from 11.63.5.2 port 49153
At time 2.02876s server sent 272 bytes to 11.63.5.2 port 49153
At time 2.03844s server received 272 bytes from 11.63.5.1 port 49153
At time 2.03844s server sent 272 bytes to 11.63.5.1 port 49153
At time 2.04895s client received 272 bytes from 11.63.5.6 port 9
At time 2.0578s client received 272 bytes from 11.63.5.3 port 9
At time 2.06735s client received 272 bytes from 11.63.5.4 port 9
At time 2.07684s client received 272 bytes from 11.63.5.5 port 9
Flow 1 (11.63.5.1 -> 11.63.5.5)
Tx Bytes: 300
Rx Bytes: 300
FlowThroughput: 0.8624295 Mbps
FlowDelay: 0.0384434 Second
Flow 2 (11.63.5.2 -> 11.63.5.6)
Tx Bytes: 300
Rx Bytes: 300
FlowThroughput: 0.8834394 Mbps
FlowDelay: 0.0287634 Second
Flow 3 (11.63.5.7 -> 11.63.5.3)
Tx Bytes: 300
Rx Bytes: 300
FlowThroughput: 0.124569 Mbps
FlowDelay: 0.0192664 Second
Flow 4 (11.63.5.6 -> 11.63.5.4)
Tx Bytes: 300
Rx Bytes: 300
FlowThroughput: 0.165661 Mbps
FlowDelay: 0.0144874 Second
Flow 5 (11.63.5.4 -> 11.63.5.8)
Tx Bytes: 300
Rx Bytes: 300
FlowThroughput: 0.0454001 Mbps
FlowDelay: 0.0528634 Second
Flow 6 (11.63.5.3 -> 11.63.5.7)
Tx Bytes: 300
Rx Bytes: 300
FlowThroughput: 0.8622853 Mbps
FlowDelay: 0.0385324 Second
Flow 7 (11.63.5.6 -> 11.63.5.2)
Tx Bytes: 300
Rx Bytes: 300
FlowThroughput: 0.124408 Mbps
FlowDelay: 0.0192914 Second
Flow 8 (11.63.5.5 -> 11.63.5.1)
Tx Bytes: 300
Rx Bytes: 300
FlowThroughput: 0.8624994 Mbps
FlowDelay: 0.0384004 Second
Total FlowThroughput: 0.109025 Mbps
Total FlowDelay one direction: 0.0312156 Second
Total FlowDelay end to end: 0.062512 Second
(base) amirhosein@amirhosein-UXSS50VD:~/ns3/ns-allinnone-3.30/ns-3.30$ ./waf --run "scratch/HW3_Non --Interval=0.034 --PacketSize=544"
waf: Entering directory '/home/amirhosein/ns3/ns-allinnone-3.30/ns-3.30/build'
waf: leaving directory '/home/amirhosein/ns3/ns-allinnone-3.30/ns-3.30/build'
```

Total Throughput: 0.109025 Mbps

Total end to end delay: 0.062512 seconds

4- دیتا ریت 128 کیلوبیت بر ثانیه:

```
Activities Terminal
amirhosein@amirhosein-UX550VD: ~/ns3/ns-allinone-3.30/ns-3.30
(base) amirhosein@amirhosein-UX550VD:~/ns3/ns-allinone-3.30/ns-3.30$ ./waf --run "scratch/HW3_Non --Interval=0.034 --PacketSize=544"
waf: Entering directory '/home/amirhosein/ns3/ns-allinone-3.30/ns-3.30/build'
waf: Leaving directory '/home/amirhosein/ns3/ns-allinone-3.30/ns-3.30/build'
Build commands will be stored in build/compile_commands.json
build finished successfully (0.722s)
At time 2s client sent 544 bytes to 11.63.5.5 port 9
At time 2s client sent 544 bytes to 11.63.5.6 port 9
At time 2s client sent 544 bytes to 11.63.5.3 port 9
At time 2s client sent 544 bytes to 11.63.5.4 port 9
At time 2.01661s server received 544 bytes from 11.63.5.8 port 49153
At time 2.01661s server sent 544 bytes to 11.63.5.8 port 49153
At time 2.03985s server sent 544 bytes to 11.63.5.2 port 49153
At time 2.02823s server sent 544 bytes to 11.63.5.2 port 49153
At time 2.03985s server received 544 bytes from 11.63.5.1 port 49153
At time 2.05156s client received 544 bytes from 11.63.5.4 port 9
At time 2.05824s server received 544 bytes from 11.63.5.7 port 49153
At time 2.05824s server sent 544 bytes to 11.63.5.7 port 49153
At time 2.07914s client received 544 bytes from 11.63.5.6 port 9
At time 2.08195s client received 544 bytes from 11.63.5.5 port 9
At time 2.09358s client received 544 bytes from 11.63.5.3 port 9
Flow 1 (11.63.5.1 -> 11.63.5.5)
Tx Bytes: 572
Rx Bytes: 572
FlowThroughput: 0.114818 Mbps
FlowDelay: 0.0398544 Second
Flow 2 (11.63.5.2 -> 11.63.5.6)
Tx Bytes: 572
Rx Bytes: 572
FlowThroughput: 0.162083 Mbps
FlowDelay: 0.0282324 Second
Flow 3 (11.63.5.7 -> 11.63.5.3)
Tx Bytes: 572
Rx Bytes: 572
FlowThroughput: 0.0785682 Mbps
FlowDelay: 0.0582424 Second
Flow 4 (11.63.5.8 -> 11.63.5.4)
Tx Bytes: 572
Rx Bytes: 572
FlowThroughput: 0.275457 Mbps
FlowDelay: 0.0166124 Second
Flow 5 (11.63.5.4 -> 11.63.5.8)
Tx Bytes: 572
Rx Bytes: 572
FlowThroughput: 0.130951 Mbps
FlowDelay: 0.0349444 Second
Flow 6 (11.63.5.6 -> 11.63.5.2)
Tx Bytes: 572
Rx Bytes: 572
FlowThroughput: 0.109191 Mbps
FlowDelay: 0.0419084 Second
Flow 7 (11.63.5.5 -> 11.63.5.1)
Tx Bytes: 572
Rx Bytes: 572
FlowThroughput: 0.100708 Mbps
FlowDelay: 0.0420944 Second
Flow 8 (11.63.5.3 -> 11.63.5.7)
Tx Bytes: 572
Rx Bytes: 572
FlowThroughput: 0.129498 Mbps
FlowDelay: 0.0353364 Second
Total FlowThroughput: 0.157732 Mbps
Total FlowDelay one direction: 0.0371531 Second
Total FlowDelay end to end: 0.0743063 Second
(base) amirhosein@amirhosein-UX550VD:~/ns3/ns-allinone-3.30/ns-3.30$ ./waf --run "scratch/HW3_Non --Interval=0.034 --PacketSize=1088"
waf: Entering directory '/home/amirhosein/ns3/ns-allinone-3.30/ns-3.30/build'
waf: Leaving directory '/home/amirhosein/ns3/ns-allinone-3.30/ns-3.30/build'
Build commands will be stored in build/compile_commands.json
build finished successfully (0.721s)
At time 2s client sent 1088 bytes to 11.63.5.5 port 9
At time 2s client sent 1088 bytes to 11.63.5.6 port 9
At time 2s client sent 1088 bytes to 11.63.5.3 port 9
At time 2s client sent 1088 bytes to 11.63.5.4 port 9
At time 2.02086s server received 1088 bytes from 11.63.5.8 port 49153
At time 2.03074s server received 1088 bytes from 11.63.5.1 port 49153
At time 2.03074s server sent 1088 bytes to 11.63.5.1 port 49153
At time 2.05262s client received 1088 bytes from 11.63.5.4 port 9
At time 2.06355s server received 1088 bytes from 11.63.5.7 port 49153
At time 2.06355s server sent 1088 bytes to 11.63.5.7 port 49153
At time 2.07996s server received 1088 bytes from 11.63.5.2 port 49153
At time 2.07996s server sent 1088 bytes to 11.63.5.2 port 49153
At time 2.0959s client received 1088 bytes from 11.63.5.5 port 9
At time 2.11214s client received 1088 bytes from 11.63.5.6 port 9
```

ادامه:

```
Activities Terminal
amirhosein@amirhosein-UX550VD: ~/ns3/ns-allinone-3.30/ns-3.30
Rx Bytes: 572
FlowThroughput: 0.162083 Mbps
FlowDelay: 0.0282324 Second
Flow 3 (11.63.5.7 -> 11.63.5.3)
Tx Bytes: 572
Rx Bytes: 572
FlowThroughput: 0.0785682 Mbps
FlowDelay: 0.0582424 Second
Flow 4 (11.63.5.8 -> 11.63.5.4)
Tx Bytes: 572
Rx Bytes: 572
FlowThroughput: 0.275457 Mbps
FlowDelay: 0.0166124 Second
Flow 5 (11.63.5.4 -> 11.63.5.8)
Tx Bytes: 572
Rx Bytes: 572
FlowThroughput: 0.130951 Mbps
FlowDelay: 0.0349444 Second
Flow 6 (11.63.5.6 -> 11.63.5.2)
Tx Bytes: 572
Rx Bytes: 572
FlowThroughput: 0.109191 Mbps
FlowDelay: 0.0419084 Second
Flow 7 (11.63.5.5 -> 11.63.5.1)
Tx Bytes: 572
Rx Bytes: 572
FlowThroughput: 0.100708 Mbps
FlowDelay: 0.0420944 Second
Flow 8 (11.63.5.3 -> 11.63.5.7)
Tx Bytes: 572
Rx Bytes: 572
FlowThroughput: 0.129498 Mbps
FlowDelay: 0.0353364 Second
Total FlowThroughput: 0.157732 Mbps
Total FlowDelay one direction: 0.0371531 Second
Total FlowDelay end to end: 0.0743063 Second
(base) amirhosein@amirhosein-UX550VD:~/ns3/ns-allinone-3.30/ns-3.30$ ./waf --run "scratch/HW3_Non --Interval=0.034 --PacketSize=1088"
waf: Entering directory '/home/amirhosein/ns3/ns-allinone-3.30/ns-3.30/build'
waf: Leaving directory '/home/amirhosein/ns3/ns-allinone-3.30/ns-3.30/build'
Build commands will be stored in build/compile_commands.json
build finished successfully (0.721s)
At time 2s client sent 1088 bytes to 11.63.5.5 port 9
At time 2s client sent 1088 bytes to 11.63.5.6 port 9
At time 2s client sent 1088 bytes to 11.63.5.3 port 9
At time 2s client sent 1088 bytes to 11.63.5.4 port 9
At time 2.02086s server received 1088 bytes from 11.63.5.8 port 49153
At time 2.03074s server received 1088 bytes from 11.63.5.1 port 49153
At time 2.03074s server sent 1088 bytes to 11.63.5.1 port 49153
At time 2.05262s client received 1088 bytes from 11.63.5.4 port 9
At time 2.06355s server received 1088 bytes from 11.63.5.7 port 49153
At time 2.06355s server sent 1088 bytes to 11.63.5.7 port 49153
At time 2.07996s server received 1088 bytes from 11.63.5.2 port 49153
At time 2.07996s server sent 1088 bytes to 11.63.5.2 port 49153
At time 2.0959s client received 1088 bytes from 11.63.5.5 port 9
At time 2.11214s client received 1088 bytes from 11.63.5.6 port 9
```

Total Throughput: 0.157732 Mbps

Total end to end delay: 0.0743063 seconds

5- دیتا ریت 256 کیلوبیت بر ثانیه:

```
Activities Terminal
amirhosein@amirhosein-UX550VD: ~/ns3/ns-allinone-3.30/ns-3.30
(base) amirhosein@amirhosein-UX550VD:~/ns3/ns-allinone-3.30/ns-3.30$ ./waf --run "scratch/HW3_Non --Interval=0.034 --PacketSize=1088"
waf: Entering directory '/home/amirhosein/ns3/ns-allinone-3.30/ns-3.30/build'
waf: Leaving directory '/home/amirhosein/ns3/ns-allinone-3.30/ns-3.30/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.731s)
At time 2s client sent 1088 bytes to 11.63.5.5 port 9
At time 2s client sent 1088 bytes to 11.63.5.6 port 9
At time 2s client sent 1088 bytes to 11.63.5.3 port 9
At time 2s client sent 1088 bytes to 11.63.5.4 port 9
At time 2.02086s server received 1088 bytes from 11.63.5.8 port 49153
At time 2.02086s server sent 1088 bytes to 11.63.5.8 port 49153
At time 2.03074s server received 1088 bytes from 11.63.5.1 port 49153
At time 2.03074s server sent 1088 bytes to 11.63.5.2 port 49153
At time 2.05262s client received 1088 bytes from 11.63.5.4 port 9
At time 2.06355s server received 1088 bytes from 11.63.5.7 port 49153
At time 2.06355s server sent 1088 bytes to 11.63.5.7 port 49153
At time 2.07996s server received 1088 bytes from 11.63.5.2 port 49153
At time 2.07996s server sent 1088 bytes to 11.63.5.2 port 49153
At time 2.0959s client received 1088 bytes from 11.63.5.5 port 9
At time 2.1121s client received 1088 bytes from 11.63.5.6 port 9
At time 2.1284s client received 1088 bytes from 11.63.5.3 port 9
Flow 1 (11.63.5.1 -> 11.63.5.5)
Tx Bytes: 1116
Rx Bytes: 1116
FlowThroughput: 0.242983 Mbps
FlowDelay: 0.0367434 Second
Flow 2 (11.63.5.2 -> 11.63.5.6)
Tx Bytes: 1116
Rx Bytes: 1116
FlowThroughput: 0.111654 Mbps
FlowDelay: 0.0799614 Second
Flow 3 (11.63.5.7 -> 11.63.5.3)
Tx Bytes: 1116
Rx Bytes: 1116
FlowThroughput: 0.140478 Mbps
FlowDelay: 0.0635544 Second
Flow 4 (11.63.5.8 -> 11.63.5.4)
Tx Bytes: 1116
Rx Bytes: 1116
FlowThroughput: 0.281105 Mbps
FlowDelay: 0.0317684 Second
Flow 5 (11.63.5.4 -> 11.63.5.8)
Tx Bytes: 1116
Rx Bytes: 1116
FlowThroughput: 0.150922 Mbps
FlowDelay: 0.0591564 Second
Flow 6 (11.63.5.5 -> 11.63.5.1)
Tx Bytes: 1116
Rx Bytes: 1116
FlowThroughput: 0.137681 Mbps
FlowDelay: 0.0648454 Second
Flow 7 (11.63.5.3 -> 11.63.5.7)
Tx Bytes: 1116
Rx Bytes: 1116
FlowThroughput: 0.137681 Mbps
FlowDelay: 0.0648454 Second
Flow 8 (11.63.5.6 -> 11.63.5.2)
Tx Bytes: 1116
Rx Bytes: 1116
FlowThroughput: 0.277488 Mbps
FlowDelay: 0.0321744 Second
Total FlowThroughput: 0.230766 Mbps
Total FlowDelay one direction: 0.0486322 Second
Total FlowDelay end to end: 0.0972645 Second
(base) amirhosein@amirhosein-UX550VD:~/ns3/ns-allinone-3.30/ns-3.30$ ./waf --run "scratch/HW3_Non --Interval=0.034 --PacketSize=2176"
waf: Entering directory '/home/amirhosein/ns3/ns-allinone-3.30/ns-3.30/build'
waf: Leaving directory '/home/amirhosein/ns3/ns-allinone-3.30/ns-3.30/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.733s)
At time 2s client sent 2176 bytes to 11.63.5.5 port 9
At time 2s client sent 2176 bytes to 11.63.5.6 port 9
At time 2s client sent 2176 bytes to 11.63.5.3 port 9
At time 2s client sent 2176 bytes to 11.63.5.4 port 9
At time 2.05865s server received 2176 bytes from 11.63.5.8 port 49153
At time 2.05865s server sent 2176 bytes to 11.63.5.8 port 49153
At time 2.05865s server received 2176 bytes from 11.63.5.1 port 49153
At time 2.05865s server sent 2176 bytes to 11.63.5.1 port 49153
At time 2.09908s server received 2176 bytes from 11.63.5.2 port 49153
At time 2.09908s server sent 2176 bytes to 11.63.5.2 port 49153
At time 2.14535s client received 2176 bytes from 11.63.5.5 port 9
At time 2.15318s server received 2176 bytes from 11.63.5.7 port 49153
At time 2.15318s server sent 2176 bytes to 11.63.5.7 port 49153
At time 2.18811s client received 2176 bytes from 11.63.5.4 port 9
At time 2.20712s client received 2176 bytes from 11.63.5.3 port 9
At time 2.21519s client received 2176 bytes from 11.63.5.6 port 9
```

ادامه ...

```
Activities Terminal
amirhosein@amirhosein-UX550VD: ~/ns3/ns-allinone-3.30/ns-3.30
(base) amirhosein@amirhosein-UX550VD:~/ns3/ns-allinone-3.30/ns-3.30$ ./waf --run "scratch/HW3_Non --Interval=0.034 --PacketSize=2176"
waf: Entering directory '/home/amirhosein/ns3/ns-allinone-3.30/ns-3.30/build'
waf: Leaving directory '/home/amirhosein/ns3/ns-allinone-3.30/ns-3.30/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.733s)
At time 2s client sent 2176 bytes to 11.63.5.5 port 9
At time 2s client sent 2176 bytes to 11.63.5.6 port 9
At time 2s client sent 2176 bytes to 11.63.5.3 port 9
At time 2s client sent 2176 bytes to 11.63.5.4 port 9
At time 2.05865s server received 2176 bytes from 11.63.5.8 port 49153
At time 2.05865s server sent 2176 bytes to 11.63.5.8 port 49153
At time 2.05865s server received 2176 bytes from 11.63.5.1 port 49153
At time 2.05865s server sent 2176 bytes to 11.63.5.1 port 49153
At time 2.09908s server received 2176 bytes from 11.63.5.2 port 49153
At time 2.09908s server sent 2176 bytes to 11.63.5.2 port 49153
At time 2.14535s client received 2176 bytes from 11.63.5.5 port 9
At time 2.15318s server received 2176 bytes from 11.63.5.7 port 49153
At time 2.15318s server sent 2176 bytes to 11.63.5.7 port 49153
At time 2.18811s client received 2176 bytes from 11.63.5.4 port 9
At time 2.20712s client received 2176 bytes from 11.63.5.3 port 9
At time 2.21519s client received 2176 bytes from 11.63.5.6 port 9
```

Total Throughput: 0.230766 Mbps

Total end to end delay: 0.0972645 seconds

6- دیتا ریت 512 کیلوبیت بر ثانیه:

```
Activities Terminal
amirhosein@amirhosein-UX550VD: ~/ns3/ns-allinone-3.30/ns-3.30
Total FlowDelay end to end: 0.0972645 Second
(base) amirhosein@amirhosein-UX550VD:~/ns3/ns-allinone-3.30/ns-3.30$ ./waf --run "scratch/HW3_Non --Interval=0.034 --Packetsize=2176"
waf: Entering directory '/home/amirhosein/ns3/ns-allinone-3.30/build'
waf: Leaving directory '/home/amirhosein/ns3/ns-allinone-3.30/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.735s)
At time 2s client sent 2176 bytes to 11.63.5.5 port 9
At time 2s client sent 2176 bytes to 11.63.5.6 port 9
At time 2s client sent 2176 bytes to 11.63.5.3 port 9
At time 2.05065s server received 2176 bytes from 11.63.5.8 port 49153
At time 2.05065s server sent 2176 bytes to 11.63.5.1 port 49153
At time 2.05065s server sent 2176 bytes to 11.63.5.1 port 49153
At time 2.09968s server sent 2176 bytes to 11.63.5.2 port 49153
At time 2.14535s client received 2176 bytes from 11.63.5.5 port 9
At time 2.15318s server received 2176 bytes from 11.63.5.7 port 49153
At time 2.15318s server sent 2176 bytes to 11.63.5.7 port 49153
At time 2.19011s client received 2176 bytes from 11.63.5.4 port 9
At time 2.20712s client received 2176 bytes from 11.63.5.3 port 9
At time 2.21518s client received 2176 bytes from 11.63.5.6 port 9
Flow 1 (11.63.5.1 -> 11.63.5.5)
Tx Bytes: 2204
Rx Bytes: 2204
FlowThroughput: 0.301113 Mbps
FlowDelay: 0.058556 Second
Flow 2 (11.63.5.2 -> 11.63.5.6)
Tx Bytes: 2204
Rx Bytes: 2204
FlowThroughput: 0.176881 Mbps
FlowDelay: 0.099683 Second
Flow 3 (11.63.5.7 -> 11.63.5.3)
Tx Bytes: 2204
Rx Bytes: 2204
FlowThroughput: 0.115103 Mbps
FlowDelay: 0.153184 Second
Flow 4 (11.63.5.8 -> 11.63.5.4)
Tx Bytes: 2204
Rx Bytes: 2204
FlowThroughput: 0.348142 Mbps
FlowDelay: 0.050646 Second
Flow 5 (11.63.5.4 -> 11.63.5.8)
Tx Bytes: 2204
Rx Bytes: 2204
FlowThroughput: 0.136194 Mbps
FlowDelay: 0.129462 Second
Flow 6 (11.63.5.5 -> 11.63.5.1)
Tx Bytes: 2204
Rx Bytes: 2204
FlowThroughput: 0.203136 Mbps
FlowDelay: 0.086799 Second
Flow 7 (11.63.5.6 -> 11.63.5.2)
Tx Bytes: 2204
Rx Bytes: 2204
FlowThroughput: 0.152646 Mbps
FlowDelay: 0.115509 Second
Flow 8 (11.63.5.3 -> 11.63.5.7)
Tx Bytes: 2204
Rx Bytes: 2204
FlowThroughput: 0.326876 Mbps
FlowDelay: 0.053941 Second
Total FlowThroughput: 0.23531 Mbps
Total FlowDelay one direction: 0.0934725 Second
Total FlowDelay end to end: 0.186945 Second
(base) amirhosein@amirhosein-UX550VD:~/ns3/ns-allinone-3.30/ns-3.30$ ./waf --run "scratch/HW3_Non --Interval=0.034 --Packetsize=4352"
waf: Entering directory '/home/amirhosein/ns3/ns-allinone-3.30/build'
waf: Leaving directory '/home/amirhosein/ns3/ns-allinone-3.30/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.735s)
At time 2s client sent 4352 bytes to 11.63.5.5 port 9
At time 2s client sent 4352 bytes to 11.63.5.6 port 9
At time 2s client sent 4352 bytes to 11.63.5.3 port 9
At time 2.11694s server received 4352 bytes from 11.63.5.8 port 49153
At time 2.11694s server sent 4352 bytes to 11.63.5.1 port 49153
At time 2.163s server received 4352 bytes from 11.63.5.1 port 49153
At time 2.163s server sent 4352 bytes to 11.63.5.1 port 49153
At time 2.19051s server sent 4352 bytes to 11.63.5.7 port 49153
At time 2.21772s server received 4352 bytes from 11.63.5.2 port 49153
At time 2.21772s server sent 4352 bytes to 11.63.5.2 port 49153
At time 2.28017s client received 4352 bytes from 11.63.5.4 port 9
At time 2.32973s client received 4352 bytes from 11.63.5.6 port 9
At time 2.3573s client received 4352 bytes from 11.63.5.3 port 9
At time 2.37899s client received 4352 bytes from 11.63.5.5 port 9
Flow 1 (11.63.5.1 -> 11.63.5.5)
Tx Bytes: 4380
```

ادامه:

```
Activities Terminal
amirhosein@amirhosein-UX550VD: ~/ns3/ns-allinone-3.30/ns-3.30
Total FlowThroughput: 0.23531 Mbps
Total FlowDelay one direction: 0.0934725 Second
Total FlowDelay end to end: 0.186945 Second
(base) amirhosein@amirhosein-UX550VD:~/ns3/ns-allinone-3.30/ns-3.30$ ./waf --run "scratch/HW3_Non --Interval=0.034 --Packetsize=4352"
waf: Entering directory '/home/amirhosein/ns3/ns-allinone-3.30/build'
waf: Leaving directory '/home/amirhosein/ns3/ns-allinone-3.30/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.735s)
At time 2s client sent 4352 bytes to 11.63.5.5 port 9
At time 2s client sent 4352 bytes to 11.63.5.6 port 9
At time 2s client sent 4352 bytes to 11.63.5.3 port 9
At time 2.11694s server received 4352 bytes from 11.63.5.8 port 49153
At time 2.11694s server sent 4352 bytes to 11.63.5.1 port 49153
At time 2.163s server received 4352 bytes from 11.63.5.1 port 49153
At time 2.163s server sent 4352 bytes to 11.63.5.1 port 49153
At time 2.19051s server sent 4352 bytes to 11.63.5.7 port 49153
At time 2.21772s server received 4352 bytes from 11.63.5.2 port 49153
At time 2.21772s server sent 4352 bytes to 11.63.5.2 port 49153
At time 2.28017s client received 4352 bytes from 11.63.5.4 port 9
At time 2.32973s client received 4352 bytes from 11.63.5.6 port 9
At time 2.3573s client received 4352 bytes from 11.63.5.3 port 9
At time 2.37899s client received 4352 bytes from 11.63.5.5 port 9
Flow 1 (11.63.5.1 -> 11.63.5.5)
Tx Bytes: 4380
```

Total Throughput: 0.23531 Mbps

Total end to end delay: 0.186945 seconds

7- دیتا ریت 512 کیلو بیت بر ثانیه:

```
Activities Terminal
amirhosein@amirhosein-UX550VD: ~/ns3/ns-allinone-3.30/ns-3.30
(base) amirhosein@amirhosein-UX550VD:~/ns3/ns-allinone-3.30/ns-3.30$ ./waf --run "scratch/HM3_Non --Interval=0.034 --PacketSize=4352"
waf: Entering directory '/home/amirhosein/ns3/ns-allinone-3.30/ns-3.30/build'
waf: Leaving directory '/home/amirhosein/ns3/ns-allinone-3.30/ns-3.30/build'
build commands will be stored in build/compile_commands.json
'build' Finished successfully (0.73ss)
At time 2s client sent 4352 bytes to 11.63.5.5 port 9
At time 2s client sent 4352 bytes to 11.63.5.6 port 9
At time 2s client sent 4352 bytes to 11.63.5.3 port 9
At time 2s client sent 4352 bytes to 11.63.5.4 port 9
At time 2.11694s server received 4352 bytes from 11.63.5.8 port 49153
At time 2.163s server received 4352 bytes from 11.63.5.1 port 49153
At time 2.19051s server sent 4352 bytes to 11.63.5.7 port 49153
At time 2.21772s server received 4352 bytes from 11.63.5.2 port 49153
At time 2.21772s server sent 4352 bytes to 11.63.5.2 port 49153
At time 2.25017s client received 4352 bytes from 11.63.5.4 port 9
At time 2.32973s client received 4352 bytes from 11.63.5.6 port 9
At time 2.3573s client received 4352 bytes from 11.63.5.3 port 9
At time 2.37099s client received 4352 bytes from 11.63.5.5 port 9
Flow 1 (11.63.5.1 -> 11.63.5.5)
Tx Bytes: 4380
Rx Bytes: 4380
FlowThroughput: 0.214963 Mbps
FlowDelay: 0.163085 Second
Flow 2 (11.63.5.2 -> 11.63.5.6)
Tx Bytes: 4380
Rx Bytes: 4380
FlowThroughput: 0.160944 Mbps
FlowDelay: 0.217716 Second
Flow 3 (11.63.5.7 -> 11.63.5.3)
Tx Bytes: 4380
Rx Bytes: 4380
FlowThroughput: 0.183926 Mbps
FlowDelay: 0.190512 Second
Flow 4 (11.63.5.8 -> 11.63.5.4)
Tx Bytes: 4380
Rx Bytes: 4380
FlowThroughput: 0.299652 Mbps
FlowDelay: 0.116936 Second
Flow 5 (11.63.5.4 -> 11.63.5.8)
Tx Bytes: 4380
Rx Bytes: 4380
FlowThroughput: 0.262995 Mbps
FlowDelay: 0.133235 Second
Flow 6 (11.63.5.5 -> 11.63.5.1)
Tx Bytes: 4380
Rx Bytes: 4380
FlowThroughput: 0.168477 Mbps
FlowDelay: 0.207981 Second
Flow 7 (11.63.5.3 -> 11.63.5.7)
Tx Bytes: 4380
Rx Bytes: 4380
FlowThroughput: 0.210082 Mbps
```

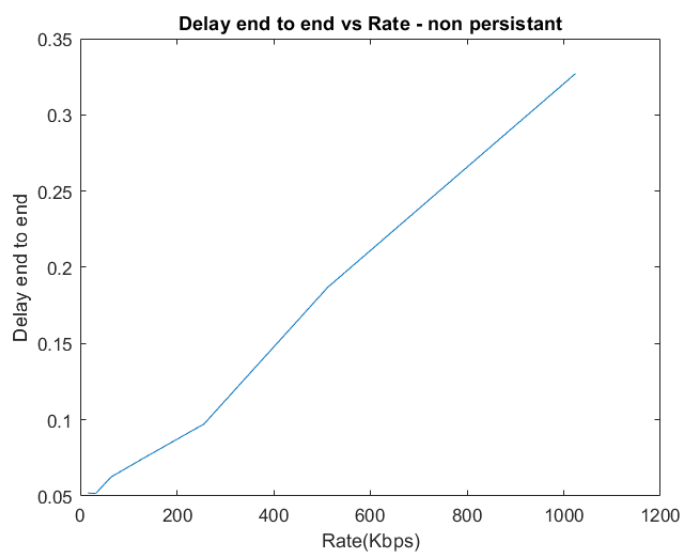
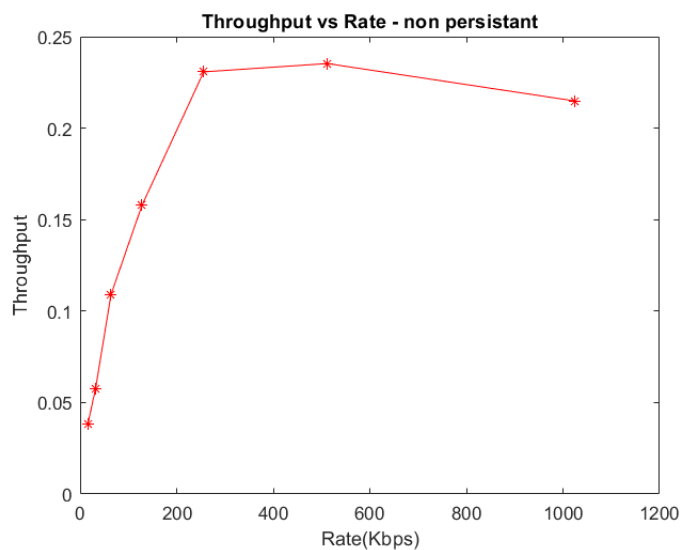
ادامه:

```
Activities Terminal
amirhosein@amirhosein-UX550VD: ~/ns3/ns-allinone-3.30/ns-3.30
At time 2.11694s server sent 4352 bytes to 11.63.5.8 port 49153
At time 2.163s server received 4352 bytes from 11.63.5.1 port 49153
At time 2.19051s server sent 4352 bytes to 11.63.5.7 port 49153
At time 2.21772s server received 4352 bytes from 11.63.5.2 port 49153
At time 2.21772s server sent 4352 bytes to 11.63.5.2 port 49153
At time 2.25017s client received 4352 bytes from 11.63.5.4 port 9
At time 2.32973s client received 4352 bytes from 11.63.5.6 port 9
At time 2.3573s client received 4352 bytes from 11.63.5.3 port 9
At time 2.37099s client received 4352 bytes from 11.63.5.5 port 9
Flow 1 (11.63.5.1 -> 11.63.5.5)
Tx Bytes: 4380
Rx Bytes: 4380
FlowThroughput: 0.214963 Mbps
FlowDelay: 0.163085 Second
Flow 2 (11.63.5.2 -> 11.63.5.6)
Tx Bytes: 4380
Rx Bytes: 4380
FlowThroughput: 0.160944 Mbps
FlowDelay: 0.217716 Second
Flow 3 (11.63.5.7 -> 11.63.5.3)
Tx Bytes: 4380
Rx Bytes: 4380
FlowThroughput: 0.183926 Mbps
FlowDelay: 0.190512 Second
Flow 4 (11.63.5.8 -> 11.63.5.4)
Tx Bytes: 4380
Rx Bytes: 4380
FlowThroughput: 0.299652 Mbps
FlowDelay: 0.116936 Second
Flow 5 (11.63.5.4 -> 11.63.5.8)
Tx Bytes: 4380
Rx Bytes: 4380
FlowThroughput: 0.262995 Mbps
FlowDelay: 0.133235 Second
Flow 6 (11.63.5.5 -> 11.63.5.1)
Tx Bytes: 4380
Rx Bytes: 4380
FlowThroughput: 0.168477 Mbps
FlowDelay: 0.207981 Second
Flow 7 (11.63.5.3 -> 11.63.5.7)
Tx Bytes: 4380
Rx Bytes: 4380
FlowThroughput: 0.210082 Mbps
Flow 8 (11.63.5.6 -> 11.63.5.2)
Tx Bytes: 4380
Rx Bytes: 4380
FlowThroughput: 0.312825 Mbps
FlowDelay: 0.112012 Second
Total FlowThroughput: 0.214871 Mbps
Total FlowDelay one direction: 0.163523 Second
Total FlowDelay end to end: 0.327046 Second
(base) amirhosein@amirhosein-UX550VD:~/ns3/ns-allinone-3.30/ns-3.30$
```

Total Throughput: 0.214871 Mbps

Total end to end delay: 0.327046 seconds

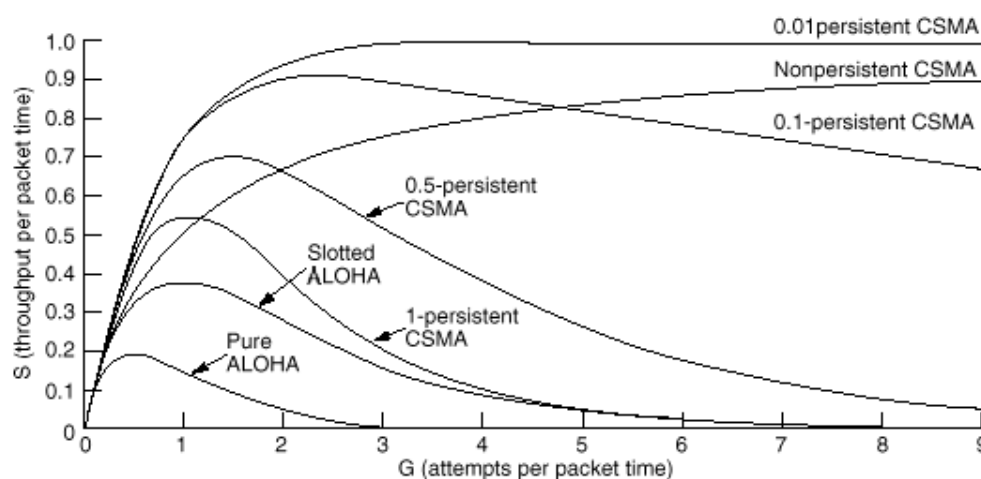
حال با کنار هم قرار دادن خروجی همه دیتا ریت ها داریم که نمودارها به شرح زیر می گردند:
نمودارهای خروجی به شکل زیر است:



Delay: از آنجایی که طول بسته زیاد می شود در ازای ریت بالاتر کاملاً منطقی است که تاخیر صعودی باشد. زیرا تمام فرستنده های دیگر زمان بیشتری باید معطل بمانند تا بسته ی برنده به مقصد برسد. به علاوه هر بسته ای که دچار collision شد چون rate زیاد می شود طول صف زیاد می شود و تاخیر شبکه افزایش می یابد.

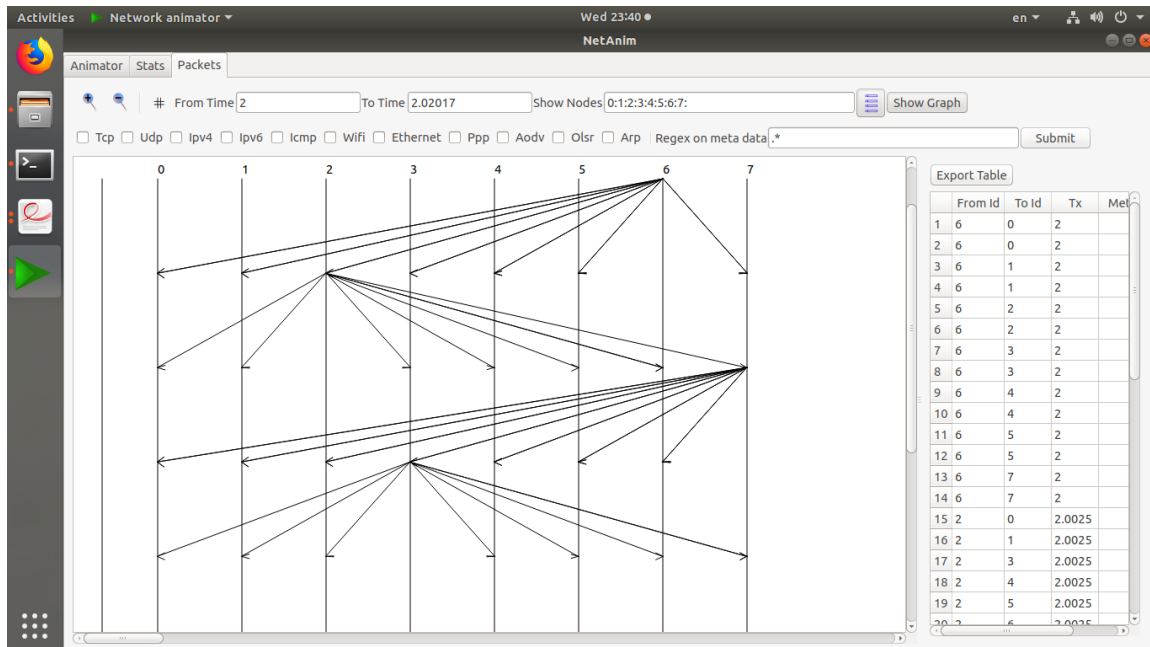
Throughput در ابتدا صعودی است زیرا با افزایش rate داده‌ی بیشتری از کانال می‌گذرد و در ابتدا کانال خالی است اما از یک حدی به بعد شیب نزولی پیدا می‌کند زیرا با اینکه طول بسته‌ها افزایش یافته ولی زمان صرف شده بیش از حد افزایش یافته است و Throughput کمی کاهش پیدا می‌کند. علت اینکه چرا در ادامه کمی حالت صعودی طور پیدا می‌کند این است که کانالی که شبیه سازی کرده ایم رفتار خوبی را برای بسته‌هایی تا ماکسیمم طول 1500 بایت ارائه می‌دهد. اگر بیشتر از این باشد کانال رفتار مشابه با کانال csma واقعی را از خود نشان نمی‌دهد و اگر بالای 4400 بایت باشد دیگر هیچ بسته‌ای را از خود عبور نمی‌دهد. در نهایت تروپوت به مقدار خاصی میل می‌کند. با توجه به این که تعداد فرستنده‌ها محدود است محاسبات تئوری موجود در اسلایدها کاملاً در این جا موضوعیت ندارد ولی استدلال‌های آن درست است. با توجه به این که در صورت مشغول بودن کانال به اندازه‌ی زمان رندوم صبر می‌کنیم، با افزایش طول پکت‌ها احتمال برخورد آن‌ها کمتر شده و کانال به صورت مفیدتر استفاده می‌شود. در ابتدا هرچه که طول بسته بیشتر می‌شود با اینکه دلیلی هم افزایش می‌یابد اما نسبت تعداد بیت‌های به درستی انتقال یافته به زمان بهبود می‌یابد. در عین حال این زمان رندوم صبر کردن موجب تاخیر است. به علاوه با افزایش طول پکت‌ها کانال توسط سایرین برای مدت طولانی‌تری اشغال می‌شود و ما برای ارسال داده باید میزان بیشتری صبر کنیم.

با توجه به اسلایدها داریم:



مشاهده می‌کنید که throughput روند **صعودی** دارد برای non persistent که در نهایت در یک مقدار خاص اشباع می‌شود مشابه آن چه در شبیه‌سازی به دست آورده ایم.

اگر فایل انیمیشن را ران کنیم خروجی به شکل زیر داریم:



اگر فرستنده ها را به ترتیبی که در انیمیشن می فرستند یادداشت کنیم خروجی به صورت زیر خواهد بود:

6,2,7,3,7,6,3,7,3,1,5,1,0,4,0,2,6,2,5,1,5,4,0,4

همانطور که مشاهده می شود دقیقا 24 بار ارسال داریم و هر فرستنده دقیقا 3 بار می فرستد. در هر بار ارسال ابتدا یکی سرور فرستاده سپس کلاینت پاسخ داده است. به علت اینکه برای گرفتن کانال رقابت وجود دارد ممکن است گاهی اوقات پاسخ کلاینت بلافاصله بعد از رسیدن بسته ی سرور فرستاده نمی شود.

در ابتدا اگر نود 7 و 3 را در نظر بگیریم، ابتدا نود 7 یک بسته ی آرپ می فرستد. نود 3 پاسخ این آرپ را با یک بسته ی آرپ دیگر که حاوی آدرسش است می فرستد. حال نود 7 بسته ی اترنت را به 3 ارسال می کند. نود 3 یک بسته ی آرپ ارسال کرده تا آدرس فرستنده را بیابد. نود 7 با یک بسته ی آرپ حاوی آدرس اش پاسخ نود 3 را داده و نود 3 ack مربوط به بسته ی اترنت گرفته شده را پس می دهد. در این فرایند نودهای دیگر درحال ارسال هستند و به همین علت دقیقا پشت سر هم اتفاق نمی افتد و با فاصله است.

پیاده سازی الگوریتم p persistent به کمک تغییر فایل csma-net-device.cc:

پیش از پرداختن به پیاده سازی ابتدا به کمک شرایط ذکر شده برای p (توسط ارقام شماره دانشجویی) برای بنده p به این شکل به دست آمد:

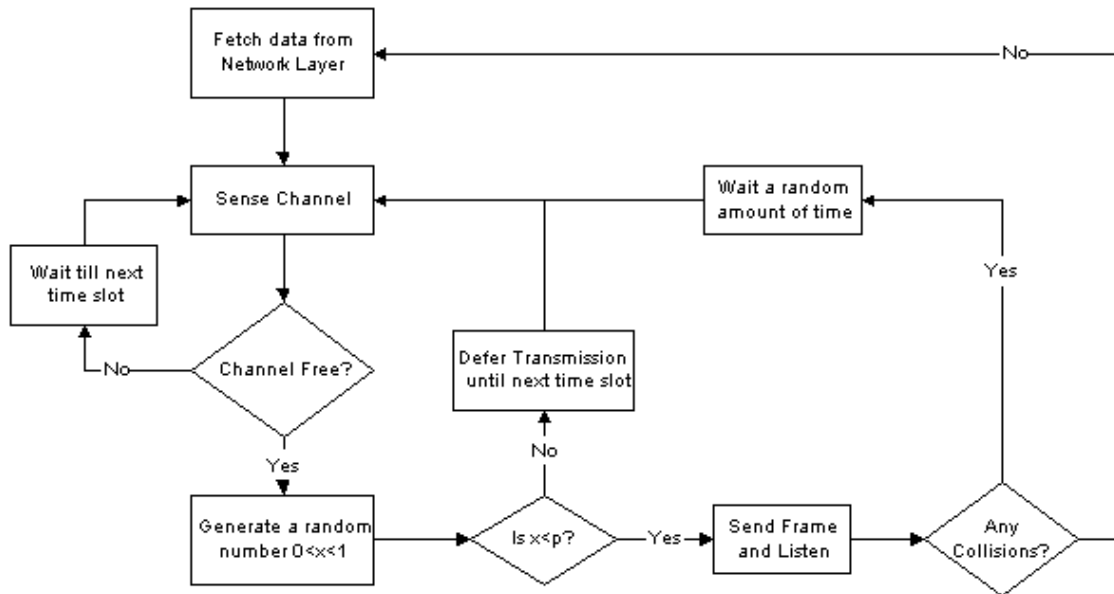
$$p_{min} = \frac{\min(f, g, h)}{10} = \frac{\min(6, 3, 5)}{10} = 0.3$$

$$p_{max} = \frac{\max(f, g, h)}{10} = \frac{\max(6, 3, 5)}{10} = 0.6$$

حال برویم سراغ پیاده سازی:

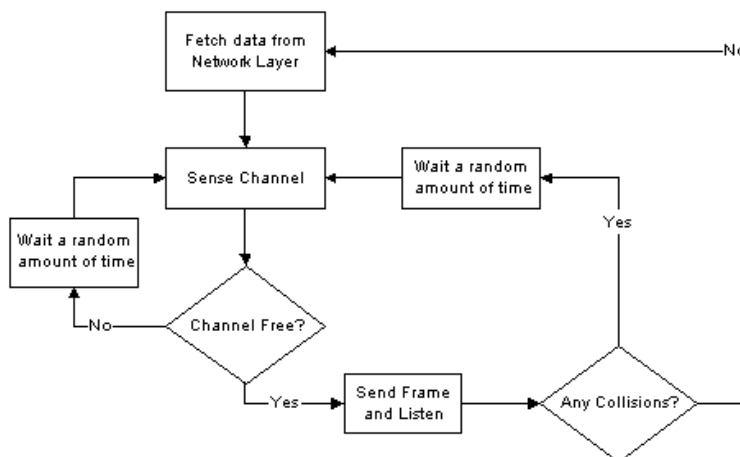
برای پیاده سازی این الگوریتم به تفاوت الگوریتم های زیر توجه کنید:

این چارت مربوط به الگوریتم p persistent است:



در صفحه ی بعد فلوچارت الگوریتم non persistent آورده شده است.

این چارت مربوط به الگوریتم non persistent است:



حال مرحله مرحله به بیان تفاوت ها و نحوه ی هندل کردن آن ها می پردازیم:

1- یکی از تفاوت هایی که در این دو الگوریتم مشاهده می کنید در زمان صبر کردن برای الگوریتم p persistent است که پس از احساس کانال به اندازه ی یک slot صبر می کند ولی در الگوریتم non persistent به اندازه ی یک زمان رندوم صبر می کند.

تغییری که در کد داده ایم به این صورت است که میزان رندوم زمان back off را به اندازه ی یک time slot تغییر داده ایم:

```

474 // if it is idle, or backoff our transmission if someone else is on the wire.
475 //
476 if (m_channel->GetState () != IDLE)
477 {
478     //
479     // The channel is busy -- backoff and rechedule TransmitStart() unless
480     // we have exhausted all of our retries.
481     //
482     m_txMachineState = BACKOFF;
483
484     if (m_backoff.MaxRetriesReached ())
485     {
486         //
487         // Too many retries, abort transmission of packet
488         //
489         TransmitAbort ();
490     }
491     else
492     {
493         m_macTxBackoffTrace (m_currentPkt);
494         m_backoff.IncrNumRetries ();
495
496         //Time backoffTime = m_backoff.GetBackoffTime (); → random delay → non persistent.
497         changed TO Time backoffTime = m_backoff.m_slotTime; // wait till next slot }→ one slot delay → p persistent.
498
499         NS_LOG_LOGIC ("Channel busy, backing off for " << backoffTime.GetSeconds () << " sec");
500         Simulator::Schedule (backoffTime, &CsmaNetDevice::TransmitStart, this);
501     }
502 }
503
504 else // the channel is free
505 {
506
507

```

2- تفاوت دیگری که این دو الگوریتم دارند این است که **اولا** اگر کانال خالی بود الگوریتم دوم به احتمال p داده را ارسال می کند **ثانیا** در صورت عدم رخداد شرط p دوباره به اندازه ی یک time slot باید صبر کنیم. با استفاده از کتابخانه های مربوط به پیاده سازی اعداد رندوم و با توجه به این که دوباره به اندازه ی یک time slot باید صبر کنیم تغییرات زیر را اعمال کردیم. به علاوه در صورتی که collision رخ داد باید به اندازه ی یک زمان رندوم باید صبر کنیم.

توجه کنید که این تغییرات برای حالت $p = 0.3$ است و برای حالت $p = 0.6$ کافی است هر جا 0.3 بود به 0.6 تغییرش بدهید.

```

510     m_rng = CreateObject<UniformRandomVariable> ();
511     float p = m_rng->GetValue (0, 1);
512
513     if (p < 0.3)  ← اعمال شرط برای ارسال
514     {
515
516         m_phyTxBeginTrace (m_currentPkt);
517
518         /*if (m_channel->TransmitStart (m_currentPkt, m_deviceId) == false)
519         {
520             NS_LOG_WARN ("Channel TransmitStart returns an error");
521             m_phyTxDropTrace (m_currentPkt);
522             m_currentPkt = 0;
523             m_txMachineState = READY;
524         }*/
525
526         if (m_channel->TransmitStart (m_currentPkt, m_deviceId) == false)
527         {
528             NS_LOG_WARN ("Channel TransmitStart returns an error");
529             m_phyTxDropTrace (m_currentPkt);
530             m_txMachineState = BACKOFF;
531
532             if (m_backoff.MaxRetriesReached ())
533             {
534                 //
535                 // Too many retries, abort transmission of packet
536                 //
537                 TransmitAbort ();
538             }
539             else
540             {
541                 m_macTxBackoffTrace (m_currentPkt);
542
543                 m_backoff.IncrNumRetries ();
544                 m_backoff.GetBackoffTime ();

```

در قسمت زیر اگر طبق اندازه‌ی p ارسال رخ نداد به اندازه‌ی یک time slot صبر می‌کنیم:

```

563 }
564 } // end of if (p < 0.3)
565 else { // (p > 0.3)
566     // set a backoff time we want to wait
567     m_txMachineState = BACKOFF;
568
569     if (m_backoff.MaxRetriesReached ())
570     {
571         //
572         // Too many retries, abort transmission of packet
573         //
574         TransmitAbort ();
575     }
576     else
577     {
578         m_macTxBackoffTrace (m_currentPkt);
579         m_backoff.IncrNumRetries ();
580         //Time backoffTime = m_backoff.GetBackoffTime ();
581         Time backoffTime = m_backoff.m_slotTime; // wait till next slot
582         NS_LOG_LOGIC ("Channel busy, backing off for " << backoffTime.GetSeconds () << " sec");
583         Simulator::Schedule (backoffTime, &CsmaNetDevice::TransmitStart, this);
584     }
585 } // end of // (p > 0.3)
586

```

در این حالت به اندازه‌ی یک slot صبر می‌کنیم.

(From non persistent.)

random

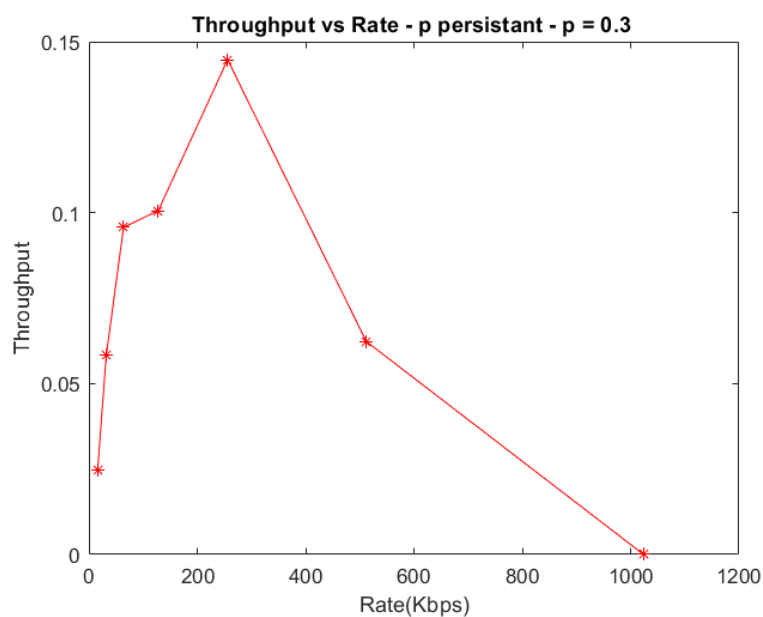
one slot delay.

changed To.

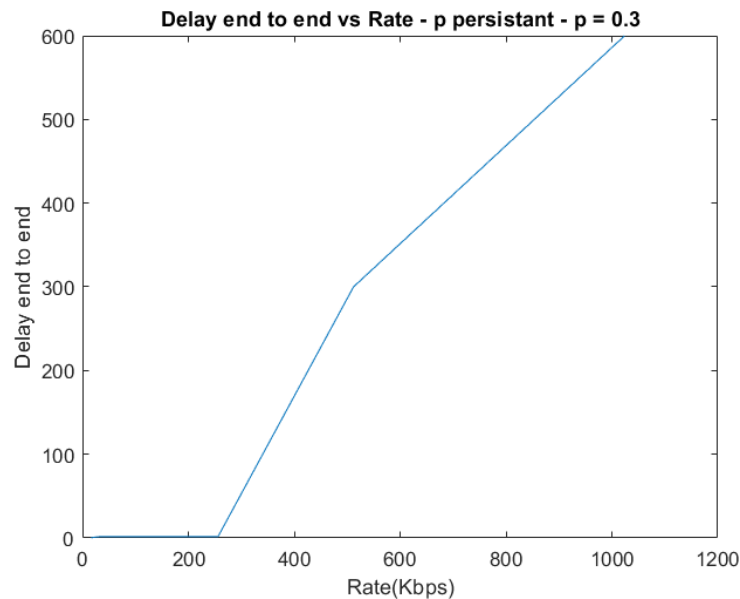
حال به بررسی نتایج به ازای دو p مختلف می‌پردازیم:

$P = 0.3$

نمودار Throughput:



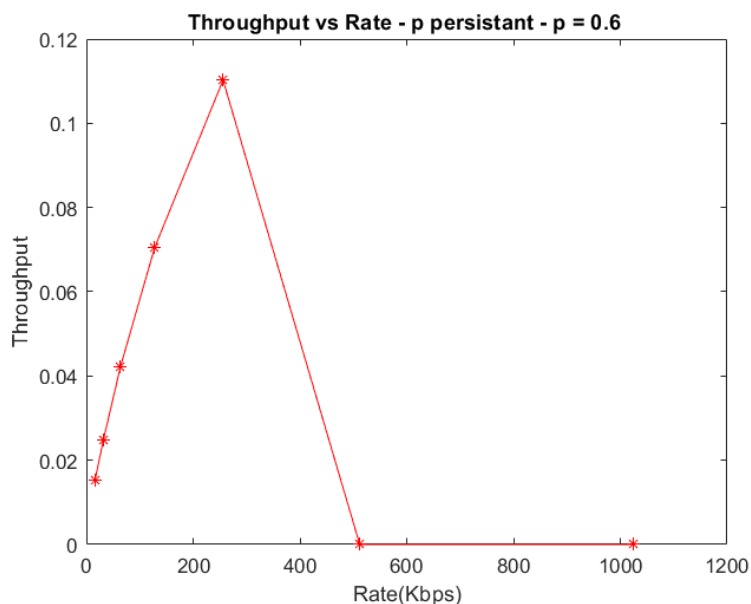
نمودار end to end delay :



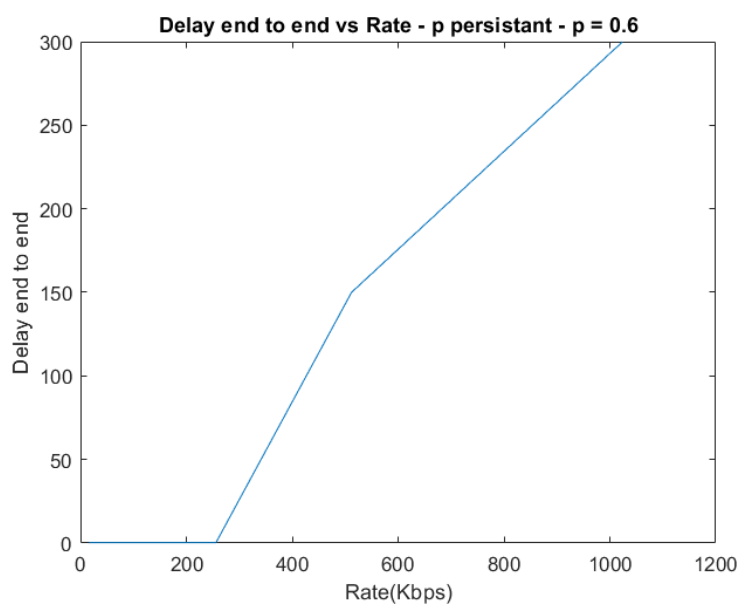
با توجه به نتایج بالا مشاهده می‌کنید که در $p = 0.3$ با افزایش نرخ داده، در ابتدا چون کانال خالی است throughput افزایش پیدا می‌کند ولی در نهایت به دلیل بالا رفتن ترافیک شبکه این مقدار کم می‌شود. استدلال‌های مربوط به non persistent در این جا نیز برای توجیه روند تغییرات throughput و delay صادق است. هر چند همانطور که بعدا نیز اشاره خواهد شد، چون با یک احتمالی داده ارسال می‌شود و ممکن است ارسال آن را به تعویق بیندازیم، کانال ممکن است خالی بماند و تروپوت نسبت به حالت قبل کمتر است. ولی این باعث می‌شود که میزان تصادم در کانال کمتر شود. از طرفی هر چقدر احتمال p کمتر باشد خواهیم داشت که با بسته‌ها با فیلتر تنگ تری رو به رو خواهند بود و این فیلترینگ باعث می‌شود throughput افزایش یابد (نسبت به حالت با p بیشتر) اما چون هرچه p کمتر شود میزان صبر کردن ما هم افزایش می‌یابد طبیعتا delay هم افزایش خواهد یافت (رابطه ی $delay$ و p معکوس است و رابطه ی مستقیم بین p و throughput).

حال برویم سراغ $P = 0.6$:

نمودار Throughput:



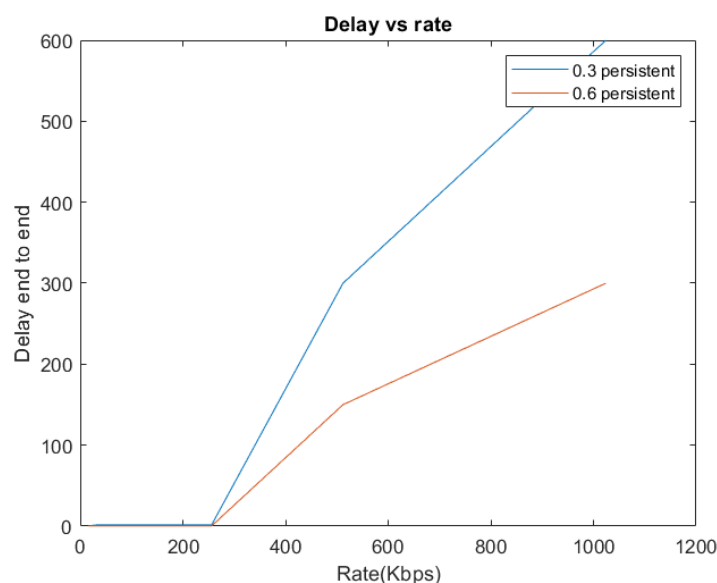
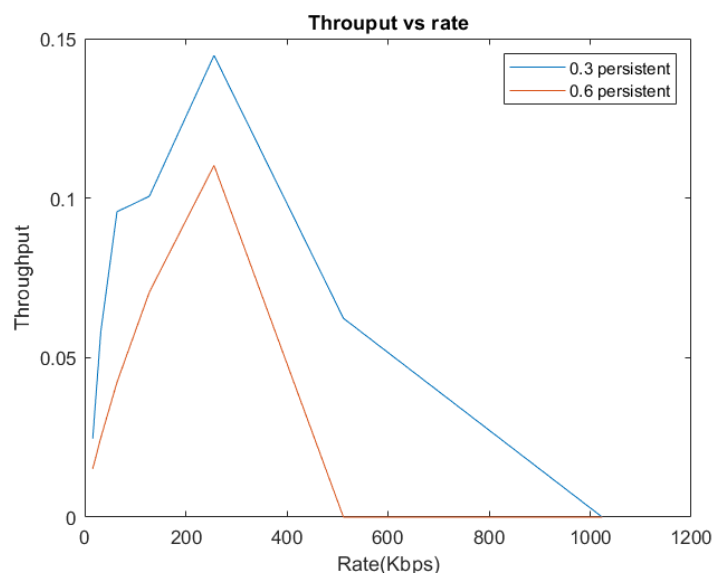
نمودار end to end delay:



با توجه به نتایج دو حالت p مشاهده می کنیم که در $p = 0.6$ با افزایش نرخ داده، در ابتدا چون کانال خالی است throughput افزایش پیدا می کند ولی در نهایت به دلیل بالا رفتن ترافیک شبکه این مقدار کم می شود، استدلال های مربوط به non persistent هم در اینجا نیز برای توجیه روند تغییرات throughput و delay صادق است. داریم که چون ارسال ها با یک احتمالی علاوه بر حالت non ارسال می شوند ممکن است ارسال ها به تعویق بیافتند و کانال

ممکن است خالی بماند و لذا throughput نسبت به حالت non کمتر است ولی این کار باعث کاهش تصادم می گردد، علی ذلک در این حالت نسبت به حالت $p = 0.3$ چون سخت گیری کمتری در ارسال/عدم ارسال بسته ها می شود لذا داریم که احتمال برخورد نیز بالا می رود و throughput در این حالت نسبت به حالت قبلی کاهش پیدا می کند. اما چون شرط فیلترینگمان بیشتر شده است صبر کمتری لازم است بکنیم و یعنی تاخیر نیز نسبت به حالت $p=0.3$ نیز کاهش پیدا می کند.

حال در ادامه نمودارهای این دو p را روی یکدیگر ملاحظه کرده و تایید روند تعوری را در آن ها ملاحظه می کنیم:



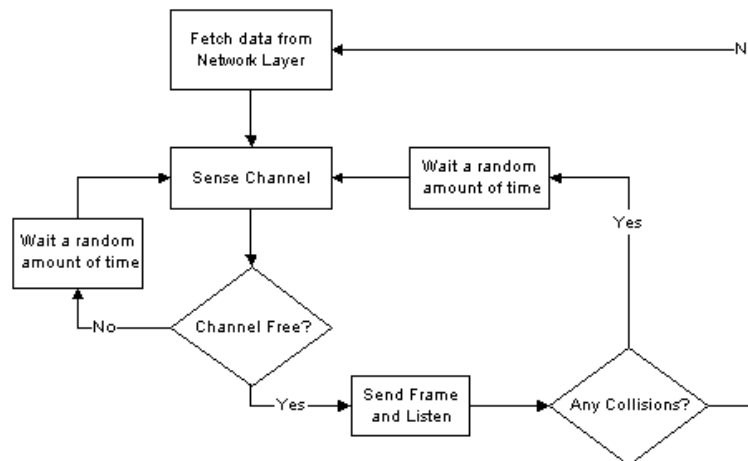
با افزایش سایز پکت ها تاخیر به شدت زیاد می شد و برخی وقت ها عددش بسیار زیاد بود (نمونه هایی از این حالات در ترمینال در ادامه آورده خواهد شد).

ذکر یک نکته کلی:

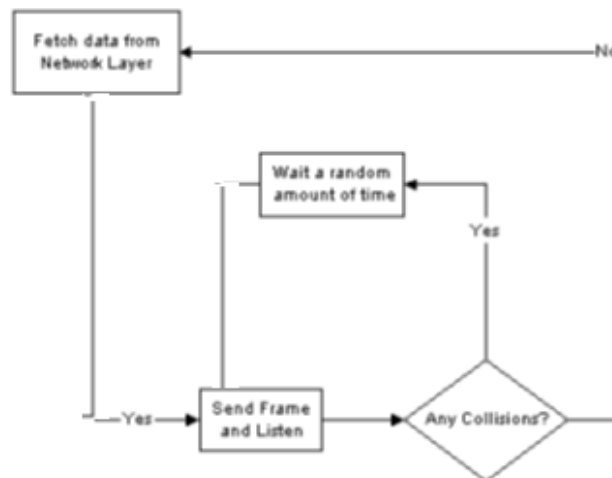
از آنجا که در هر کدام از این لایه های مک از "احتمال" استفاده می کنیم، در حالت تست با دفعات محدود گاهی وقت ها برخی حالت ها خوش شانس تر از بقیه ی حالت ها خروجی می دهند و این به خاطر استفاده از عدد رندوم در کد، رخ می دهد، لذا اگر در گزارش و فایل های tr. و فایل های pcap. در برخی حالت ها اندکی تفاوت در قیاس با حالت تعوری وجود داشت به معنای غلط بودن آن نیست بلکه به خاطر ذات احتمالی تست های ما و محدود بودن تست هایمان این اتفاق رخ داده است اما بنده با تکرار تست ها به اندازه ی کافی سعی کردم از رخداد این قسم وقایع پیشگیری کنم و در کنار وجود این قسم خطا ها، هم چنان روند کلی هر کدام از الگوریتم های لایه مک مطابق با حالت نظری اش است (همانطور که ملاحظه کردید در نمودار های بخش های گذشته، روند کلی نمودار ها نکات مربوطه ی تعوری را تایید می کرد).

پیاده سازی الگوریتم ALOHA به کمک تغییر فایل csma-net-device.cc:

به فلوچارت های پایین دقت کنید:



Non persistent



ALOHA

با توجه به تفاوت ها در الگوریتم های بالا تغییرات زیر را در کد اعمال کردیم:

1- ابتدا تمام قسمت‌های مربوط به sense کردن کانال را حذف کردیم:

```

472 //-----
473 // In ALOHA the sender always sends the packet, no matter what.
474 // This part of the code should be commented. Because in pure ALOHA the channel is not sensed.
475 //-----
476
477 // Now we have to sense the state of the medium and either start transmitting
478 // if it is idle, or backoff our transmission if someone else is on the wire.
479 //
480
481 /*
482 if (m_channel->GetState () != IDLE)
483 {
484     //
485     // The channel is busy -- backoff and rechedule TransmitStart() unless
486     // we have exhausted all of our retries.
487     //
488     m_txMachineState = BACKOFF;
489
490     if (m_backoff.MaxRetriesReached ())
491     {
492         //
493         // Too many retries, abort transmission of packet
494         //
495         TransmitAbort ();
496     }
497     else
498     {
499         m_macTxBackoffTrace (m_currentPkt);
500
501         m_backoff.IncrNumRetries ();
502         Time backoffTime = m_backoff.GetBackoffTime ();

```

توی بهرست کانال

حذف بررسی وضعیت کانال
commenting channel sense.

2- هر گاه بسته‌ای برای ارسال داشتیم، ارسال می‌کنیم. در هنگام ارسال چنانچه collision رخ داد به اندازه‌ی یک تایم رندوم صبر می‌کنیم:

```

509 */
510 // else
511 // {
512 //     //
513 //     The channel is free, transmit the packet
514 //     //
515     m_phyTxBeginTrace (m_currentPkt);
516     if (m_channel->TransmitStart (m_currentPkt, m_deviceId) == false)
517     {
518         NS_LOG_WARN ("Channel TransmitStart returns an error");
519         // m_phyTxDropTrace (m_currentPkt);
520         m_txMachineState = BACKOFF;
521
522         if (m_backoff.MaxRetriesReached ())
523         {
524             //
525             // Too many retries, abort transmission of packet
526             //
527             TransmitAbort ();
528         }
529         else
530         {
531             m_macTxBackoffTrace (m_currentPkt);
532
533             m_backoff.IncrNumRetries ();
534             Time backoffTime = m_backoff.GetBackoffTime ();
535
536             NS_LOG_LOGIC ("Collision, backing off for " << backoffTime.GetSeconds () << " sec");
537
538             Simulator::Schedule (backoffTime, &CsmaNetDevice::TransmitStart, this);
539         }

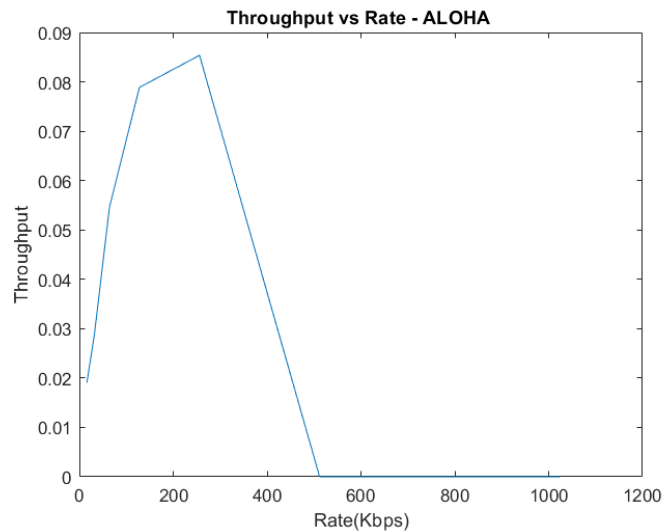
```

→ set random delay time.

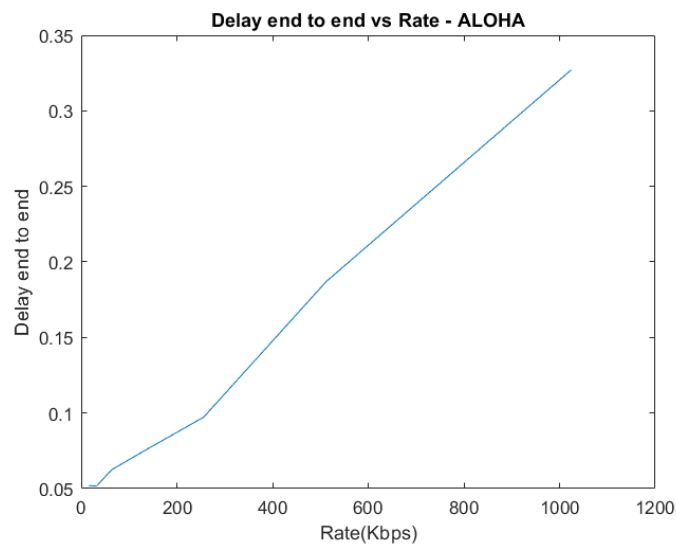
در کد بالا دقیقاً آن کاری که هنگام idle نبودن کانال در non persistent انجام می‌دادیم، اینجا در هنگام عدم موفقیت ارسال انجام می‌دهیم.

نمودارهای حاصل:

نمودار Throughput:



نمودار end to end delay:



در تحلیل ALOHA ، همانطور که در نمودارهای بالا مشاهده می کنید با افزایش سایز پکت ها تاخیر در شبکه زیاد می شود. دلیل نیز این است که با افزایش rate در شبکه احتمال برخورد در بین پیام ها افزایش می یابد و تاخیر زیاد می شود. طبق فرمول زیر داریم:

$$E[T_{ALOHA}] = X + t_{prop} + (e^{2G} - 1)(X + 2t_{prop} + B)$$

با اضافه کردن G یا اضافه کردن X به شدت تاخیر شبکه زیاد می‌شود. (در این حالت با افزایش اندازه‌ی پکت، X را زیاد کرده ایم.)

خروجی زیر به ازای سایز پکت بزرگتر است:

```
At time 2s client sent 5000 bytes to 12.3.6.5 port 9
At time 2s client sent 5000 bytes to 12.3.6.6 port 9
At time 2s client sent 5000 bytes to 12.3.6.3 port 9
At time 2s client sent 5000 bytes to 12.3.6.4 port 9
Flow 1 (12.3.6.1 -> 12.3.6.5)
Tx Bytes: 5028
Rx Bytes: 0
Throughput: -0 Mbps
Delay: -2 Second
Flow 2 (12.3.6.2 -> 12.3.6.6)
Tx Bytes: 5028
Rx Bytes: 0
Throughput: -0 Mbps
Delay: -2 Second
Flow 3 (12.3.6.7 -> 12.3.6.3)
Tx Bytes: 5028
Rx Bytes: 0
Throughput: -0 Mbps
Delay: -2 Second
Flow 4 (12.3.6.8 -> 12.3.6.4)
Tx Bytes: 5028
Rx Bytes: 0
Throughput: -0 Mbps
Delay: -2 Second
Total Throughput: -0 Mbps
Total Delay one direction: 4.49358e+13 Second
Total Delay end to end: 8.98716e+13 Second
```

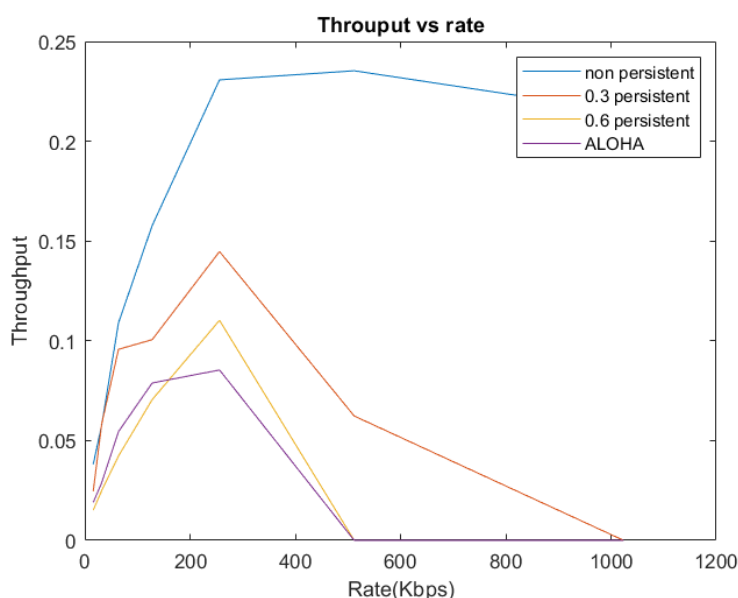
مشاهده میکنید که تاخیر بسیار زیاد و throughput 0 شده است.

در ALOHA با افزایش نرخ، در ابتدا چون کانال خالی است تروپوت زیاد می‌شود ولی در نهایت به دلیل شلوغ شدن کانال با نرخ نمایی به صفر میکند چون میزان برخوردها زیاد شده و به علاوه چون سایز پکت زیاد است با هر بار برخورد زمان بیشتری در کانال اتلاف می‌شود. تاخیر آن نسبت به روش p persistent کمتر است زیرا بسته‌ها به سرعت ارسال می‌شوند ولی تروپوت آن کمتر است.

مقایسه:

: Throughput

مشاهده می‌کنید که در الگوریتم‌ها non persistent بر خلاف الگوریتم‌های p persistent و ALOHA، با افزایش سایز پکت از یک حدی به بعد، throughput حدوداً اشباع می‌شود. ولی برای دو الگوریتم دیگر ذکر شده سرعت میل به صفر سریع‌تر است. مشاهده می‌کنید که بهترین حالت سایر الگوریتم‌ها بهتر از ALOHA هستند زیرا به شرایط کانال هنگام ارسال پکت توجه می‌کنند. با کاهش p در p persistent میزان تروپوت بهبود می‌یابد زیرا با کم احتمال کردن ارسال پکت، احتمال برخورد و در نتیجه نیاز به صبر کردن‌های طولانی برای ارسال مجدد کاهش می‌یابد. اختلاف روش ALOHA با 0.6 persistent زیاد نیست از نظر تروپوت. نمودار تروپوت الگوریتم‌ها در شکل زیر نشان داده شده است.

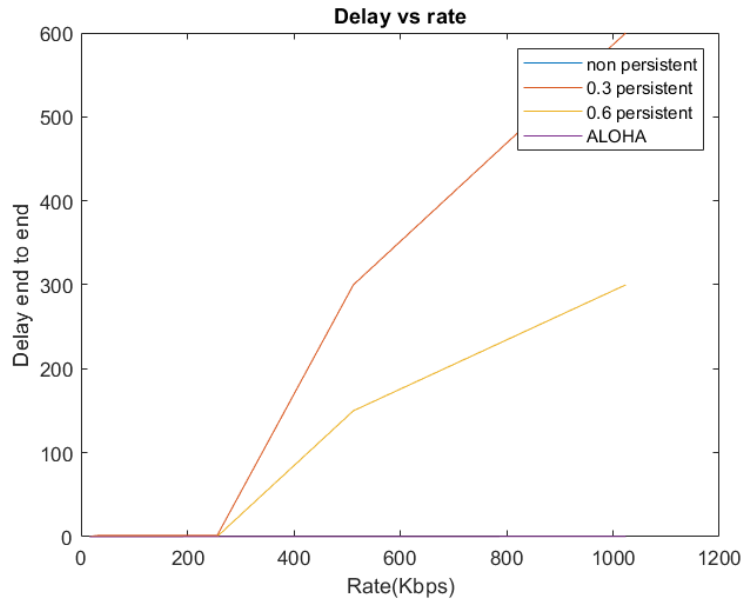


:Delay

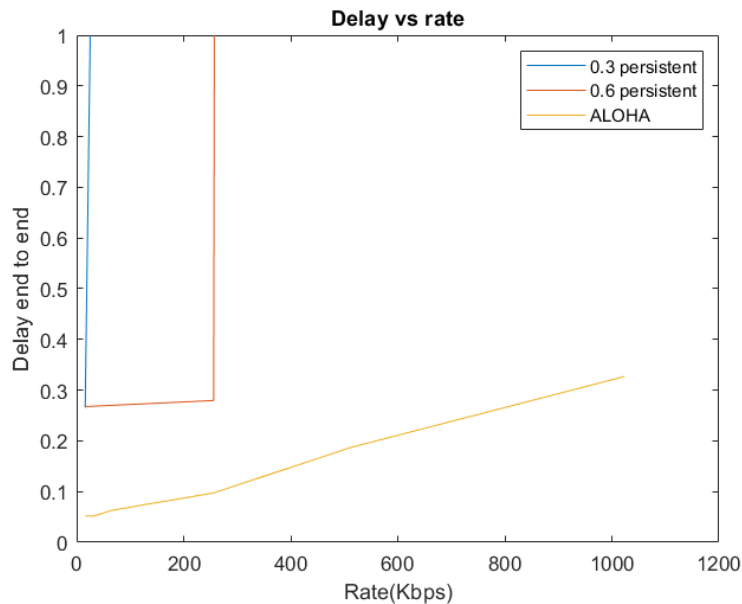
میزان دیلی در ALOHA در ریت‌های پایین زیاد نیست، زیرا به محض دریافت پکت آن را ارسال می‌کند ولی در الگوریتم‌های دیگر با کاهش p میزان تاخیر افزایش می‌یابد زیرا با احتمال کمتری داده را ارسال می‌کند و به صورت مداوم نیز کانال چک می‌شود که به تاخیر بیشتر منجر می‌شود.

در واقع بین دیلی و تروپوت در این الگوریتم‌ها یک trade off وجود دارد.

نمودار دلیلی ها در صفحه ی بعد آورده شده است.



توجه کنید که تاخیر ALOHA صفر نیست! بلکه در قیاس با تاخیر های دیگر کوچک دیده شده است، برای رفع ابهام در بازه ی 0 تا 100 زوم می کنیم:



توجه کنید که شیب 90 p persistent درجه نیست بلکه به علت رشد شدید آن در قیاس با ALOHA به صورت قایم دیده می شود حال آنکه اگر نمودار قبلی را مشاهده کنید خواهید دید که این شیب ها قایم نیستند.

توجه کنید که نمودارها به دلیل این که ماهیتا به عوامل احتمالی مربوط است کمی ممکن است با حالت تئوری متفاوت باشد.